

AMENDMENT OF THE CLAIMS

1. (Currently Amended) A method for enhancing persistence of a message, the method comprising:
storing the message in an inbound queue after receiving the message;
browsing ~~[[an]]~~the inbound queue to identify the message after storing the message in the
inbound queue;
copying the message to a working queue, the working queue being persisted by a queue manager, to persist the message, the message being in both the inbound queue and the working queue concurrently ~~before the message is removed from the inbound queue;~~
removing the message from the inbound queue after copying the message to the working queue; [[and]]
processing the message to generate a reply prior to removing the message from the working queue; and
storing the reply in an outbound queue after generating the reply.
2. (Currently Amended) The method of claim 1, further comprising removing the message from the working queue after storing the reply in ~~[[an]]~~the outbound queue.
3. (Original) The method of claim 1, further comprising restoring the message in the working queue after a system failure.
4. (Original) The method of claim 1, further comprising determining that the message is persisted prior to removing the message from the inbound queue.
5. (Currently Amended) The method of claim 1, wherein browsing comprises searching the working queue for the message as part of a wave of messages in a chronologically adjacent order to facilitate generation of the reply, wherein the message is waiting to be processed.

6. (Original) The method of claim 1, wherein browsing comprises locking the message until the message is copied to the working queue.
7. (Original) The method of claim 1, wherein processing comprises assigning the message to a thread, the thread being available to process the message.
8. (Original) The method of claim 1, wherein processing comprises transmitting a second message to request data indicated by a content of the message and generating the reply based upon data received in response to the second message.
9. (Currently Amended) An apparatus for enhancing persistence of a message, the apparatus comprising:
 - an inbound queue to receive the message from a requestor;
 - a working queue to store the message;
 - an outbound queue to store a reply responsive to the message;
 - a queue manager to persist the message from the inbound queue and the working queue before the message is removed from the inbound queue; and
 - a dispatcher to browse the inbound queue to identify the message after the message is stored in the inbound queue;[[,]] copy the message to the working queue to persist the message, the message to be in both the inbound queue and the working queue concurrently;[[,]] remove the message from the inbound queue after the message is copied to the working queue and after the message is persisted from the working queue;[[,]] and assign a thread to process the message[[,]] to generate[[ing a]]the reply in response to the message prior to removing the message from the working queue and to store the reply in the outbound queue after generating the reply.
10. (Currently Amended) The apparatus of claim 9, ~~further comprising anywherein the~~ outbound queue is to store the reply until the reply is transmitted to the requestor.
11. (Original) The apparatus of claim 10, wherein the queue manager is configured to persist the message from the inbound queue and the reply from the outbound queue.

12. (Currently Amended) The apparatus of claim 9, wherein the dispatcher comprises a persistence determiner coupled with the queue manager to determine that the message is persisted prior to removing the message from the inbound queue, wherein the dispatcher is coupled with the working queue to wait until the thread is available to assign to the message or the thread is being cleaned, before copying the message to the working queue.
13. (Currently Amended) The apparatus of claim 9, wherein the dispatcher comprises a queue searcher to identify the message to be processed based upon priorities associated with messages stored in the inbound queue.
14. (Original) The apparatus of claim 9, wherein the dispatcher comprises a message locker to lock the message, until the message is copied into the working queue.
15. (Original) The apparatus of claim 9, wherein the dispatcher comprises recovery logic to assign the thread to process the message after a system failure.
16. (Original) The apparatus of claim 9, wherein the thread is configured to process the message based upon a rule associated with the message.
17. (Currently Amended) A storage-type machine-accessible medium containing instructions, which when executed by a machine, cause said machine to perform operations, comprising:
storing the message in an inbound queue after receiving the message;
browsing [[an]]the inbound queue to identify the message after storing the message in the inbound queue;
copying the message to a working queue, the working queue being persisted by a queue manager, to persist the message, the message being in both the inbound queue and the working queue concurrently before the message is removed from the inbound queue;
removing the message from the inbound queue after copying the message to the working queue; [[and]]

processing the message to generate a reply prior to removing the message from the working queue; and
storing the reply in an outbound queue after generating the reply.

18. (Currently Amended) The storage-type machine-accessible medium of claim 17, wherein the operations further comprise removing the message from the working queue after storing the reply in an outbound queue.
19. (Currently Amended) The storage-type machine-accessible medium of claim 17, wherein the operations further comprise restoring the message in the working queue after a system failure based upon an order or priority associated with the message with respect to other messages in the working queue.
20. (Currently Amended) The storage-type machine-accessible medium of claim 17, wherein browsing comprises selecting a set of messages, the message being part of the set.